Task Blocking
Survey of methods used
Wed, Feb 18, 1998

The software in local stations/IRMs consists of 14 tasks, each of which blocks upon certain conditions. The conditions used are either reading from a message queue or waiting for a task-triggering event. Task events are supported by the pSOS kernel. Every task has a sent of 13 events that are its own. (Think of these as bits in a word.) Any task can send an event, or a set of events, to a target task. Any task can wait upon a set of events, any of which can unblock the task that is waiting. If an event is sent to a task more than once, without allowing the CPU to schedule the task, the effect is the same as setting a bit more than once. This note describes the methods used by each task for blocking. Most, maybe all, tasks are written as initialization code followed by entry into an infinite loop, at the top of which the code either waits for a message queue or waits for a set of events. In the following table, the E/Q column indicates whether the task waits on a message queue or an event mask. If it waits on an event mask, then each of the events handled by that task is described. Note that there are cases in which a task handles an event on which it does not block.

Task	E/Q	Meaning
Classic	Clas	Classic protocol datagram received with or w/o IP
Alarm	0019	
	0001	Perform alarm scan on data pool
	0008	Closed loops processing (obsolete)
	0010	Process D0 alarm message received
Console	0003	
	0001	Process mode switches, keyboard char received
	0002	Process unit switches, knob counter received
Application	0038	
	0020	Re-enable auto-page scan
	0010	Call page appl with network trigger
	0008	Call page appl with 15 Hz cycle trigger
	0004	Call page appl with keyboard interrupt trigger
	0002	Call page appl with terminate trigger
	0001	Call page appl with initialize trigger
Date/Time	000C	
	0008	Usual 15 Hz cycle, accumulate time-of-day.
	0004	Keyboard interrupt. Read time from screen.
	0002	Send time read from screen to all nodes via network.
SMDmp	000C	
	0008	Usual 15 Hz cycle. Update 8-byte display.
	0004	Keyboard interrupt. Accept new address or set word.

Update	`000E`	
	`0008`	Fulfill new requests, immediate first time reply.
	`0004`	Flush network queues to network.
	`0002`	Usual 15 Hz cycle. Process DAT, fulfill active requests.

QMonitor	`0003`	
	`0001`	Perform usual 15 Hz monitoring functions with timeout
	`0002`	Perform OUTPQ monitoring only w/o timeout

Server	`0002`	
	`0002`	Fulfill all server data requests due this cycle

Serial	`0002`	
	`0002`	Check and process NBS clock serial input
	`0000`	Check for local application registered to receive serial input

Acnet	`ANET`	Acnet protocol dispatching
DZero	`DREQ`	DZero protocol
AcReq	`ACRQ`	Acnet protocols
SNAP	`SNAP`	Internet protocols handler (IP, ICMP, ARP, UDP, IGMP)